A=SIT Technology
E-Government Innovationszentrum

24. März 2021 Studie · Mobile Sicherheit · E-Government · 33 Min. Lesezeit

# Sicherheitsanalyse mobiler Anwendungen für Krypto-Währungen

**Archiviert.** Dieser Inhalt wird nicht mehr aktualisiert. Er bleibt zur Information verfügbar.

**Stand** 2021-03-19
**Quelle** https://technology2.a-sit.at/veroeffentlichungen/sicherheitsanalyse-mobiler-anwendungen-fuer-krypto-waehrungen/

Zur Verwaltung virtueller Währungen wie Bitcoin oder Ethereum greifen viele NutzerInnen auf Mobilanwendungen zurück, die eine komfortable Form des Handels, An- und Verkaufs sowie weitere Dienstleistungen versprechen. Oftmals geht aus Anwendungsbeschreibungen nicht hervor, mit welchen persönlichen Daten diese Apps interagieren, wohin sie gesendet und wo bzw. wie sie gespeichert werden. Insbesondere in Ermangelung gesetzlicher Vorgaben oder technischer Richtlinien ist fraglich, ob das umgesetzte Sicherheitsniveau in der Praxis vergleichbar ist mit jenem, dem das etablierte Bankenwesen im Angebot mobiler Apps unterliegt.

Im Zuge dieses Projekts sollen im Rahmen einer Fallstudie exemplarisch ausgewählte Mobilanwendungen zur Verwaltung von Krypto-Währungen analysiert und auf ihre Sicherheit hin überprüft werden. Das Ziel besteht darin, einen Eindruck über das Sicherheitsniveau dieser Apps zu gewinnen und empirisch zu erheben, mit welchem Maß an Sorgfalt kritische bzw. personenbezogene Daten behandelt werden.

## 1. Einleitung

Die Kurswerte virtueller Währungen wie Bitcoin, Ethereum oder Tether zeugen von hoher Volatilität, signifikantem Investitionsrisiko und unausgereifter technischer sowie rechtlicher Regulierung. Um das Portfolio stets im Blick zu behalten und möglichst unmittelbar auf Kursschwankungen mittels Kauf oder Verkauf sog. „crypto coins“ reagieren zu können, eignen sich bevorzugt Applikationen für Mobilgeräte, die mit Android bzw. iOS betrieben werden. Die permanente Internetverbindung von Smartphones ermöglicht den Handel von Fiatgeld nahezu in Echtzeit und stellt durch die Übermittlung von „Push-Benachrichtigungen“ zudem sicher, dass Händler über alle relevanten Informationen zeitnah informiert werden. Mobile Betriebssysteme wie Android und iOS kontrollieren dabei den Zugriff von Apps auf sensible

Ressourcen und definieren über Schnittstellen sichere Rahmenbedingungen für Anwendungen, etwa um kryptographische Schlüssel abzulegen, Benachrichtigungen zu empfangen oder verschlüsselte Verbindungen mittels TLS aufzubauen.

Im Hinblick auf den Einsatz von Mobilanwendungen zur Verwaltung virtueller Währungen, sind die dahinterstehenden Anbieter gefordert, stets am aktuellen Stand der Technik zu operieren und die von Betriebssystemen über Schnittstellen angebotenen Sicherheitsfunktionen auch tatsächlich in ihren Applikationen einzusetzen. Erschwert wird dies in der Praxis vor allem durch die große Vielfalt heterogen ausgestatteter Endgeräte und unterschiedlicher Schnittstellen in den verschiedenen Versionen von Betriebssystemen. Sieht ein Prozess in einer App beispielsweise die Autorisierung über einen biometrischen Faktor vor, ist von Herstellerseite her abzuwägen, auf welchen Geräten und mit welchen Schnittstellen eine sichere Umsetzung möglich ist. Unter Android ist es beispielsweise erst seit Version 6 möglich, den Fingerabdruck-Scanner in Anwendungen einzusetzen. Dies impliziert, dass diese Funktionalität zum aktuellen Zeitpunkt ca. 20% der verwendeten Android-Geräte nicht zur Verfügung steht [1] . Darüber hinaus sind Hersteller von Mobilgeräten gefordert, ihre Produkte mit der entsprechenden Hardware auszustatten. Soll eine App ungeachtet dessen beliebige (mitunter veraltete) Android-Versionen unterstützen, müssen Einschränkungen beim Funktionsumfang vorgenommen oder vergleichbar sicherere Lösungen als Alternative eingesetzt werden, um das durch den biometrischen Faktor erreichbare Sicherheitsniveau nicht zu untergraben.

Um die Dynamik bei der Entwicklung von Schnittstellen zu veranschaulichen, sei erwähnt, dass mit Android 9 über eine Anpassung der entsprechenden API die Möglichkeit geschaffen wurde, auch weitere biometrische Hardware anzusprechen. Mit Android 10 wurde die bisherige API abgelöst durch eine neue, die App-Herstellern fortan die Möglichkeit einräumt, zuerst den Fingerabdruck abzufragen und gegebenenfalls stattdessen eine PIN- bzw. Passworteingabe zuzulassen [2] . In Android 11 wurde die Schnittstelle abermals umgebaut, um nach außen hin überprüfbar zu machen, wie stark bzw. schwach die konkrete Authentifizierung in Relation zur Verfügbarkeit alternativer biometrischer Methoden ist [3] . Gerade im Hinblick auf den Handel mit Krypto-Währungen bzw. der Freigabe von Transaktionen auf Mobilgeräten spielen derartige Schnittstellen eine essentielle Rolle. Eine Einschätzung, welche Implikationen hierbei laufende Änderungen und die Absenz von Features auf die Sicherheit von Anwendung haben, bedingt eine Risikoanalyse.

Zur Verwaltung virtueller Währungen interagieren entsprechende Mobilanwendungen im Regelfall mit einem „Wallet“, in dem kryptographisches Schlüsselmaterial abgelegt wird, das zum Zugriff auf die „Geldbörse“ mit der virtuellen Währung benötigt wird. In Ermangelung rechtsverbindlicher Sicherheitsvorgaben, steht es Anbietern von Apps frei, wo sie Schlüssel ablegen. Die technisch sicherste Variante ist die Verwahrung in einem „Hardware Storage“, der vom Betriebssystem nur eingeschränkt zugänglich ist. Je nach Anwendungsszenario können diese Schlüssel aber auch auf einem Server („Web Wallet“) oder in einer Datei („Software Wallet“) liegen. Je nachdem welcher Zweck von einer Anwendung angestrebt wird, lässt sich selbstverständlich keine pauschale Aussage darüber treffen, wie sicher bzw. unsicher eine gewisse Herangehensweise ist. Nichts desto trotz ermöglicht die Analyse der Implementierungssicherheit von Anwendungen einen objektiven Blick darauf, ob gegebene Sicherheitsfunktionen korrekt umgesetzt und berücksichtigt wurden.

1 https://developer.android.com/about/dashboards 2 https://android-developers.googleblog.com/2019/10/one-biometric-api-over-all-android.html 3 https://android-developers.googleblog.com/2020/09/lockscreen-and-authentication.html Die nachfolgende Studie konzentriert sich auf Android-Anwendungen, die die Verwaltung bzw. den An- und Verkauf von Krypto-Währungen erlauben. Je nach Applikation werden dazu persönliche Daten verarbeitet bzw. mit kryptographischem Schlüsselmaterial hantiert. Ohne Berücksichtigung bzw. Kenntnis der jeweiligen Risikoszenarien sollen die in ausgewählten Applikationen umgesetzten Sicherheitsmaßnahmen einer Evaluierung nach objektiven Kriterien unterzogen werden. Das Ziel ist kein bewertender Vergleich einzelner Apps untereinander, sondern eine Untersuchung, welche Herangehensweisen im Kontext von Mobilanwendungen für Krypto-Währungen vorzufinden sind, wie bzw. ob sie sicher implementiert sind und wenn faktisch abschätzbar, welche Folgen sich daraus für die praktische Verwendung einer Anwendung ergeben.

Für diesen Zweck richtet diese Studie das Augenmerk vor allem auf jene Prozesse in den untersuchten Apps, die sensible Daten, wie etwa kryptographische Schlüssel, verarbeiten, kritische System-Berechtigungen anfordern, sowie auf sicherheitsrelevanten Schutzmechanismen, wie z.B. biometrische Authentifizierung, zurückgreifen. Das Ziel besteht folglich primär in der Beantwortung der nachfolgenden Fragestellungen:

- *Welche „kritischen" Berechtigungen erfordert eine Anwendung während der Ausführung?* Als relevant gelten in diesem Zusammenhang jene Berechtigungen, die Apps den Zugriff auf persönliche Daten, wie etwa Kontakte, Konten oder Bilder, von BenutzerInnen einräumen [4] .
- *Mit welchen Servern kommuniziert eine App und wie ist dieser Datentransfer geschützt?* Diese Fragestellung zielt etwa auf die korrekte Validierung von TLS-Zertifikaten und der Verwendung von „Certificate Pinning" ab.
- *Wie interagieren Apps mit dem Betriebssystem und anderen Anwendungen?* Hierbei gilt es festzustellen, ob Apps womöglich unbeabsichtigt durch Dritt-Apps zu Handlungen instruiert (manipuliert) werden können, die nicht beabsichtigt sind.
- *Wird über das Betriebssystem auf Hardware-Schutzmechanismen zugegriffen?* Diese Fragestellung umfasst etwa die Zugriffskontrolle auf kryptographische Schlüssel.

In einem zweiten Schritt sollen für jede Anwendung folgenden Kriterien eruiert werden:

- *Möglicherweise sensible Daten, die von Apps am Mobilgerät abgelegt werden*
- *Verwendet eine App „Tracker", die Rückschlüsse auf einzelne AnwenderInnen erlauben?*

Für ein besseres Verständnis des Sicherheitskontexts der untersuchten Anwendungen wird zuletzt eine grundsätzliche Code-Analyse der Apps vorgenommen. „Reverse Engineering" der Apps ermöglicht es, ein technisch fundiertes Bild über die zuvor angeführten Fragestellungen zu erlangen und nachzuvollziehen, wie sicherheitsrelevante Konzepte implementiert wurden. Darüber hinaus eignet sich diese Form der Analyse beispielsweise dazu, folgende Kriterien zu untersuchen:

- *Implementierung von Maßnahmen um modifizierte („ge-rootete") Endgeräte zu erkennen*

- *Zugriffe auf Schnittstellen, um kryptographische Schlüssel abzulegen bzw. abzurufen*
- *Enthaltene URLs und ggf. im Code hinterlegte Zugriffstokens*
- *Verwendung kryptographischer Algorithmen und Authentifizierungsdaten*
- *Ausgabe („Leaks") sensibler Informationen in Log-Dateien*

 Provenienz von kryptographischem Schlüsselmaterial 4
https://developer.android.com/guide/topics/permissions/overview#dangerous_permissions

## 1.1. Analysierte Apps

Für die nachfolgend vorgenommene Untersuchung von Mobilanwendungen wurden Applikationen ausgewählt, die eine „Wallet"-Funktionalität anbieten bzw. den Handel von virtuellen Währungen ermöglichen. Angesichts der Vielzahl von angebotenen Apps in diesem Bereich, musste für die in diesem Projekt vorgenommene Analyse eine Abgrenzung getroffen werden. In der Annahme, dass potentielle Sicherheitsprobleme in häufig installierten Apps auch vergleichsweise viele User betreffen würden, wurden nur Anwendungen ausgewählt, die mindestens 500.000 Installationen verzeichnen und in den letzten Wochen aktualisiert worden sind. Insbesondere das letztgenannte Kriterium trägt der eingangs beschriebenen Tatsache Rechnung, dass sich Schnittstellen in mobilen Betriebssystemen laufend ändern und eine fortlaufende Anpassung durch App-Hersteller somit unumgänglich ist.

Angesichts der Tatsache, dass Implementierungen sicherheitskritischer Anwendungen für Android ein größeres Feld an Eventualitäten abdecken müssen, fokussiert sich die nachfolgende Fallstudie auf Android-Anwendungen. Da sich das Sicherheitsmodell von Android signifikant von iOS unterscheidet, können Schwächen und Schlussfolgerungen in Bezug auf die Implementierung der Android-App naturgemäß nicht auf die jeweilige iOS-App umgelegt werden.

Für eine sicherheitskritische Untersuchung wurden die folgenden, frei erhältlichen Anwendungen aus dem Google PlayStore herangezogen:

- Coinbase - Bitcoin Wallet *(10.000.000+ Installationen)* https://play.google.com/store/apps/details?id=com.coinbase.android Version 9.8.0 vom 23.01.2021
- Robinhood - Investment & Trading, Commission-free *(10.000.000+ Installationen)* https://play.google.com/store/apps/details?id=com.robinhood.android Version: 4.46.6 vom 31.01.2021
- Luno: Buy Bitcoin, Ethereum and Cryptocurrency *(5.000.000+ Installationen)* https://play.google.com/store/apps/details?id=co.bitx.android.wallet Version: 7.5.0 vom 29.01.2021
- eToro: Social-Trading *(5.000.000+ Installationen)* https://play.google.com/store/apps/details?id=com.etoro.openbook Version 296.0.0 vom 12.01.2021
- Bitcoin Wallet *(5.000.000+ Installationen)* https://play.google.com/store/apps/details?id=de.schildbach.wallet Version 8.0.7 vom 05.12.2020

- BUX X - Mobile Trading App *(1.000.000+ Installationen)* https://play.google.com/store/apps/details?id=com.getbux.android Version 2.39 vom 25.01.2021
- Abra Bitcoin Crypto Wallet Buy Trade Earn Interest *(500.000+ Installationen)* https://play.google.com/store/apps/details?id=com.plutus.wallet Version 90.0 vom 19.01.2021
- CEX.IO Cryptocurrency Exchange *(500.000+ Installationen)* https://play.google.com/store/apps/details?id=io.cex.app.prod Version 5.23.1 vom 18.01.2021

## 2. App: Coinbase – Bitcoin Wallet

einer Anwendung enthalten.

Die vorliegende Anwendung unterstützt Android

| Coinbase – Bitcoin Wallet | Col2 |
|---|---|
| **Link:** | Play Store |
| **Paket-Name:** | com.coinbase.android |
| **Version:** | 9.8.0 |
| **Veröffentlichung:** | 23.01.2021 |
| **Downloads:** | 10.000.000+ |
| **Minimal unterstützte OS-Version:** | 6.0 |

ab Version 6.0. Google schreibt Herstellern ab dieser Version vor, dass Hardware-Verschlüsselung des Geräts standardmäßig aktiviert sein muss (siehe Android CDD [5], Abschnitt 9.9).

**Executive Summary**

- Die App „Coinbase – Bitcoin Wallet“ verzeichnet laut Google Play mehr als 10 Millionen Installationen, ist für Geräte ab Android 6.0 verfügbar und verlangt bei Ausführung der jeweiligen Aktionen Zugriff auf die Kamera, den externen Speicher und Kontakte.
- Die App sieht die Verwendung eines Fingerabdruck-Scanners für den Kauf, Verkauf, das Senden von „crypto coins“, sowie zum Öffnen der Anwendung vor. Alternativ sind diese Handlungen über einen PIN möglich. Für Push-Benachrichtigungen wird der Dienst Firebase von Google verwendet und zur Absicherung der Kommunikation mit Zahlungsdienstleistern wird „Certificate Pinning“ eingesetzt.
- Bei der statischen Analyse wurden keine wesentlichen Implementierungsschwächen festgestellt.

## 2.1. Verwendung von Hardware-Schutzmechanismen

Die App setzt zumindest Android 6 (API Level 23) voraus und ersucht um Gewährung der Berechtigungen „USE_FINGERPRINT“ und „USE_BIOMETRIC“, um auf den Fingerabdruck-Scanner zuzugreifen.

Für die Generierung und Ablage von kryptographischen Schlüsseln wird auf die in Android inkludierte „Java Cryptographic Architecture“ (JCA) zurückgegriffen. Durch Angabe des KeyStore-Providers „AndroidKeyStore“ delegiert die Anwendung dem Betriebssystem die Verantwortung über die sichere Ablage. Es wurden im Kontext dieser Analyse keine Anzeichen dafür vorgefunden, dass die Anwendung sicherstellen würde, dass die Speicherung tatsächlich in einem Hardware- und nicht in einem Software-Schlüsselspeicher stattfindet. Da die App erst ab Android 6.0 betrieben werden kann und für Neugeräte ab dieser Version oben genannte Anforderung nach einem sicheren Für den Handel mit virtuellen Währungen, dem Senden von Coins an Kontakte, sowie zur Freigabe der Funktionalität nach Öffnen der Anwendung, kann wahlweise ein biometrischer Faktor verwendet werden. Alternativ wird auf einen PIN-Code zurückgegriffen, der bei der Einrichtung der Anwendung festgelegt werden muss und auch als zweiter Faktor für Aktionen dient, die andernfalls der Fingerabdruck erfüllt. Für die Implementierung der Fingerabdruck-Funktionalität wird auf eine von Google bereitgestellte Programmbibliothek zurückgegriffen [6] . Für gewisse Samsung-Geräte (z.B. Galaxy S8, S9, A20 und weitere) sind Ausnahmen bei der Verwendung hinterlegt, da es dem 5 https://source.android.com/compatibility/android-cdd.pdf 6 https://developer.android.com/jetpack/androidx/releases/biometric

## 2.2. Push Notifications

Die Anwendung verwendet für Push Notifications den Dienst Firebase von Google und bezieht PushBenachrichtigungen von der Cloud-Datenbank https://friendly-slate-746.firebaseio.com.

## 2.3. Kritische Berechtigungen

Die „Coinbase – Bitcoin Wallet“-App fordert bei Verwendung die Gewährung folgender „dangerous permissions“ an, die Zugriff auf persönliche Daten von BenutzerInnen oder sensible

- *CAMERA*
- *READ_CONTACTS* Ermöglicht der Applikation auf die gespeicherten Kontakte des Nutzers zuzugreifen.
- *READ_EXTERNAL_STORAGE* Ermöglicht es einer App von externem Speicher, etwa dem integrierten Massenspeicher oder
- *WRITE_EXTERNAL_STORAGE* Ermöglicht es einer App auf externen Speicher, etwa den integrierten Massenspeicher oder

## 2.4. Verwendung externer Frameworks für „User Tracking“

Die Beschreibung der Anwendung auf Google Play verweist auf eine Datenschutzerklärung [7] . Darin vorgesehen ist auch eine Verwendung persönlicher Daten für Zwecke, die nicht unmittelbar für die Funktionalität der Applikation erforderlich sind. Ein konkreter Bezug darauf, welche Drittdienste die Daten verarbeiten, findet sich darin nicht.

Im Code der Anwendung ist die Einbindung folgender Frameworks von Drittanbietern ersichtlich:

| Tracker | Zweck |
|---|---|
| AppsFlyer | Analyse der App-Verwendung |
| Bugsnag | Analyse und Meldung von Bugs |
| Google AdMob | Werbung |
| Google CrashLytics | Analyse und Meldung von Bugs |
| Google Firebase Analytics | Analyse der App-Verwendung |

## 2.5. Netzwerkkommunikation

Neben den im vorigen Punkt angeführten Tracking-Diensten, kommuniziert die App vorwiegend mit https://api.coinbase.com. Für Zahlungen bzw. den Kauf virtueller Währungen wird eine Schnittstelle der Firma Braintree eingebunden, die unter https://www.braintreepayments.com erreichbar ist. Damit verbunden findet auch Kommunikation mit https://checkout.paypal.com statt. Die Anwendung inkludiert eine dedizierte Konfiguration [8], die Angaben zur Sicherheit von Netzwerkkommunikation trifft. Darin ist explizit festgelegt, dass kein Verkehr über HTTP stattfinden darf, außer es betrifft die im Internet nicht gerouteten IP-Adressen 10.0.2.2 und 10.0.3.2. Wie aus dem Code der Anwendung hervor geht, betreibt der Hersteller auf diesen Adressen auf Port 3000 im internen Netzwerk eine Testversion der Braintree-Schnittstelle für Zahlungen.

7 https://www.coinbase.com/legal/privacy 8 https://developer.android.com/training/articles/security-config

### 2.5.1. Certificate Pinning

Die Applikation sieht die Verwendung bestimmter Zertifikate („Certificate Pinning“) für eine Kommunikation mit braintreepayments.com und checkout.paypal.com vor. Für Netzwerkverkehr zu sonstigen Domains (neben Trackern auch https://api.coinbase.com) konnte kein Verweis auf Pinning festgestellt werden.

## 2.6. Verwendung von Kryptographie

Die “Coinbase – Bitcoin Wallet”-Anwendung setzt Kryptographie für die verschlüsselte Ablage von kryptographischem Schlüsselmaterial, sowie zur Speicherung von Userdaten in den sog. „SharedPreferences“ ein.

Die Anwendung verwendet Kryptographie zur Ablage von kryptographischem Schlüsselmaterial in den „SharedPreferences“ der App. Bei erstmaliger Verwendung wird hierfür ein Schlüssel für die Verwendung mit AES/CBC/PKCS7Padding über ein im Gerät integriertes Hardware-Element erzeugt (Bezeichnung: MASTER_ENCRYPTION_KEY). Auch der benötigte Initialisierungsvektor (IV) für die Verwendung mit CBC wird vom System generiert Etwaige Daten, die folglich in den „SharedPrefences“ abgelegt werden, werden zunächst verschlüsselt, wobei der IV an die Daten angehängt wird. Da der verwendete Schlüssel an das Endgerät gebunden ist, besteht kein praktisches Risiko, dass die im privaten Ordner der App abgelegten Daten bei Entwendung missbräuchlich verwendet werden könnten.

Neben der Ablage von Schlüsselmaterial in den „SharedPreferences“ werden in Einstellungen dort noch weitere Attribute abgelegt. Ausgangspunkt ist hierbei ein vom User eingegebener PIN, der mit einer selbst implementierten PBKDF2-Schlüsselableitungsfunktion und einem kryptographisch sicher generiertem Salt-Wert (Verwendung der `SecureRandom` API) in 1200 Iterationen verarbeitet wird, um einen Schlüssel für AES-128 abzuleiten. Die PBKDF2-Implementierung wird dabei als Systemschnittstelle hinzugefügt und ersetzt dabei die von Android von Haus inkludierte Schlüsselableitung mittels PBKDF2. Ein Wert von zumindest 1000 Iterationen gilt gemäß PKCS#5-Spezifikation Version 2.1 (RFC 8018 [9] ) von 2017 als sicher. Aufgrund des hohen Risikos für Fehler bei der Umsetzung implementieren nur weniger Anwendungen PBKDF2 selbst. Im konkreten Fall konnten jedoch keine Mängel beobachtet werden.

Über den vom PIN mittels PBKDF2 abgeleiteten AES-Schlüssel werden folglich mittels AES/CBC/PKCS5Padding folgende Attribute verschlüsselt in den „SharedPreferences“ abgelegt:

1. account_pin
2. account_salt
3. pin_enabled_app_open
4. pin_enabled_send_money
5. biometric_enabled *6.* incoorect_pin_tries *(sic!)* 9 https://tools.ietf.org/html/rfc8018

## 3. App: Robinhood - Investment & Trading, Commission-free

als funktional erweitertes Frontend für die Bedienung der Online-Plattform.

| **Robinhood - Investment & Trading, Commission-free** | **Col2** |
|---|---|
| **Link:** | Play Store |
| **Paket-Name:** | com.robinhood.android |
| **Version:** | 4.46.6 |
| **Veröffentlichung:** | 31.01.2021 |
| **Downloads:** | 10.000.000+ |
| **Minimal unterstützte OS-Version:** | 6.0 |

Die vorliegende Anwendung unterstützt Android ab Version 6.0. Google schreibt Herstellern ab dieser Version vor, dass Hardware-Verschlüsselung des Geräts standardmäßig aktiviert sein muss (siehe Android CDD [10], Abschnitt 9.9).

**Executive Summary**

- Die App „Robinhood - Investment & Trading, Commission-free“ verzeichnet laut Google Play mehr als 10 Millionen Installationen, ist für Geräte ab Android 6.0 verfügbar und verlangt bei Ausführung der jeweiligen Aktionen Zugriff auf die Kamera, den externen Speicher, Standort des Endgeräts und Accounts zu eingerichteten Diensten.
- Die App sieht die Verwendung eines Fingerabdruck-Scanners nur für das Öffnen der Anwendung vor. Alternativ sind diese Handlungen über einen PIN möglich. Für Push-Benachrichtigungen wird der Dienst Firebase von Google verwendet.
- Die Anwendung verwendet kein „Certificate Pinning“ zur Absicherung der Kommunikation.
- Userdaten werden in den „SharedPreferences“ der App verschlüsselt abgelegt, jedoch mit einem Schlüssel und Initialisierungsvektor, die im Code der Anwendung statisch hinterlegt sind.

## 3.1. Verwendung von Hardware-Schutzmechanismen

Die App setzt zumindest Android 6 (API Level 23) voraus und ersucht um Gewährung der Berechtigungen „USE_FINGERPRINT“ und „USE_BIOMETRIC“, um auf den Fingerabdruck-Scanner zuzugreifen.

Für die Generierung und Ablage von kryptographischen Schlüsseln wird auf die in Android inkludierte „Java Cryptographic Architecture“ (JCA) zurückgegriffen. Durch Angabe des KeyStore-Providers „AndroidKeyStore“ delegiert die Anwendung dem Betriebssystem die Verantwortung über die sichere Ablage. Es wurden im Kontext dieser Analyse keine Anzeichen dafür vorgefunden, dass die Anwendung sicherstellen würde, dass die Speicherung tatsächlich in einem Hardware- und nicht in einem Software-Schlüsselspeicher stattfindet. Da die App erst ab Android 6.0 betrieben werden kann und für Neugeräte ab dieser Version oben genannte Anforderung nach einem sicheren Für das Öffnen der Anwendung kann wahlweise ein biometrischer Faktor oder ein PIN verwendet werden. Für die Implementierung der Fingerabdruck-Funktionalität wird auf eine von Google bereitgestellte Programmbibliothek zurückgegriffen [11] . Für gewisse Samsung-Geräte (z.B. Galaxy S8, S9, A20 und weitere) sind Ausnahmen bei der Verwendung hinterlegt, da es dem Vernehmen hier nach zu Schwierigkeiten kommen kann.

10 https://source.android.com/compatibility/android-cdd.pdf 11 https://developer.android.com/jetpack/androidx/releases/biometric

## 3.2. Push Notifications

Die Anwendung verwendet für Push Notifications den Dienst Firebase von Google und bezieht Notifications von https://robinhood-gcm-prod.firebaseio.com. Benachrichtigungen werden für „Alarme“ eingesetzt, etwa bei Erreichen eines gewissen Kurswertes.

## 3.3. Kritische Berechtigungen

Die „Robinhood - Investment & Trading, Commission-free“-App fordert bei Verwendung die Gewährung folgender „dangerous permissions“ an, die Zugriff auf persönliche Daten von BenutzerInnen oder sensible Geräteinformationen involvieren:

- ACCESS_BACKGROUND_LOCATION Die Anwendung kann auf den aktuellen Standort des Mobilgeräts auch zugreifen, wenn sie im Hintergrund läuft.
- ACCESS_COARSE_LOCATION Benötigt, um auf den ungefähren Standort des Geräts zuzugreifen.
- ACCESS_FINE_LOCATION Erforderlich, um auf den exakten Standort des Geräts zuzugreifen.
- *CAMERA* Erforderlich, um auf die Kamera des Mobilgeräts zugreifen zu können.
- *GET_ACCOUNTS*
- *WRITE_EXTERNAL_STORAGE* Ermöglicht es einer App auf externen Speicher, etwa den integrierten Massenspeicher oder

## 3.4. Verwendung externer Frameworks für „User Tracking“

Die Beschreibung der Anwendung auf Google Play verweist auf eine Datenschutzerklärung [12] . Darin vorgesehen ist auch eine Verwendung persönlicher Daten für Zwecke, die nicht unmittelbar für die Funktionalität der Applikation erforderlich sind. Die Datenschutzerklärung führt auch explizit die jeweiligen Dienste und den Umfang der Daten an, der von ihnen erhoben wird. Eine Ausnahme (keine Erwähnung) dazu bildet Dienst „Branch“, welcher die IP-Adresse, einen Fingerprint des aktuellen Geräts, sowie weitere gerätespezifische Merkmale an den Dienstleister sendet.

Im Code der Anwendung ist die Einbindung folgender Frameworks von Drittanbietern ersichtlich:

| Tracker | Zweck |
|---|---|
| Adjust | Marketing-Optimierung |
| Branch | Analyse der App-Verwendung |
| Google Analytics | Analyse der App-Verwendung |
| Google CrashLytics | Analyse und Meldung von Bugs |
| Google Firebase Analytics | Analyse der App-Verwendung |
| Plaid | Marketing-Optimierung |

Aus Datenschutzperspektive mitunter bedenklich ist der Einsatz der Dienste „Adjust“ und „Plaid“, die den Zweck verfolgen, Informationen über BenutzerInnen aus verschiedensten Quellen zu aggregieren und miteinander zu verknüpfen, um optimierte Marketing-Profile daraus zu erstellen.

12 https://robinhood.com/l/privacy

## 3.5. Netzwerkkommunikation

Neben den im vorigen Punkt angeführten Tracking-Diensten, kommuniziert die App vorwiegend mit https://robinhood.com bzw. Subdomains davon. Die Anwendung inkludiert keine dedizierte Konfiguration [13], die Angaben zur Sicherheit von Netzwerkkommunikation trifft.

#### 3.5.1. Certificate Pinning

Die vorliegende Applikation sieht keine Verwendung bestimmter Zertifikate („Certificate Pinning“) für

### 3.6. Verwendung von Kryptographie

Die “Robinhood - Investment & Trading, Commission-free”-Anwendung setzt Kryptographie für die verschlüsselte Speicherung von Userdaten in den sog. „SharedPreferences“ ein und sieht prinzipiell auch die Ablage von Daten in einem geschützten „Vault“ vor.

Für die Ablage beliebiger Daten in den „SharedPreferences“ implementiert die App eine Verschlüsselung mittels AES/CBC/PKCS5Padding. Hierfür werden ein Schlüssel und ein Initialisierungsvektor verwendet, die konstant im Code der Anwendung hinterlegt sind. Als Basis für beide Werte wird die Zeichenfolge `hNnS24Qi3b3n8n2owu12EYxrp2ckMvrXL4TVwPzf` herangezogen. Für den Schlüssel wird der SHA-256 Hashwert dieser Zeichenfolge verwendet, als IV der unveränderte Wert. Die Implementierung selbst findet sich in der Klasse `com.robinhood.prefs.LegacySecureStringPreference`, wobei sich annehmen ließe, dass der Ausdruck „legacy“ auf ein veraltetes Verfahren hindeutet, das von einem neueren abgelöst worden wäre. Tatsächlich kommt das Verfahren aber nach wie vor zum Einsatz.

Die Anwendung implementiert darüber hinaus die verschlüsselte Ablage beliebiger Daten in einem sog. „Vault“. Für die dabei realisierte Funktionalität wird auf einen AES-Schlüssel zurückgegriffen, der von einem sicheren Hardware-Element generiert wird.

Offenbar aus Gründen der Kompatibilität zu früheren Versionen der Anwendung existiert in der Klasse `com.robinhood.vault.internal.LegacyVaultWorkerV1` auch eine alternative Implementierung, die dasselbe (kryptographisch gebrochene bzw. unsichere) Schema umsetzt, das für die Ablage von Daten in den „SharedPreferences“ eingesetzt wird. Demzufolge wird bei Erkennen einer alten Version des „Vaults“ ebendieser entschlüsselt und in weiterer Folge mit der sichereren Verschlüsselungsvariante, basierend auf einem Schlüssel in Hardware, neu verschlüsselt.

13 https://developer.android.com/training/articles/security-config

## 4. App: Luno: Buy Bitcoin, Ethereum and Cryptocurrency

Kursänderungen zu verfolgen bzw. sich ggf. darüber benachrichtigen zu lassen.

| **Luno: Buy Bitcoin, Ethereum and Cryptocurrency** | **Col2** |
|---|---|
| **Link:** | Play Store |
| **Paket-Name:** | co.bitx.android.wallet |
| **Version:** | 7.5.0 |
| **Veröffentlichung:** | 29.01.2021 |
| **Downloads:** | 5.000.000+ |
| **Minimal unterstützte OS-Version:** | 4.1 |

Die vorliegende Anwendung unterstützt Android ab Version 4.1. Da die Schnittstelle zum Zugriff auf den Fingerabdruck-Scanner erst mit Android 6.0 eingeführt wurde, ist eine biometrische Authentifizierung auf älteren Geräten technisch nicht möglich.

**Executive Summary**

- Die App „Luno: Buy Bitcoin, Ethereum and Cryptocurrency“ verzeichnet laut Google Play mehr als 5 Millionen Installationen, ist für Geräte ab Android 4.1 verfügbar und verlangt bei Ausführung der jeweiligen Aktionen Zugriff auf die Kamera, Kontakte, den externen Speicher, das Mikrofon und Accounts zu eingerichteten Diensten.
- Die App sieht die Verwendung eines Fingerabdruck-Scanners für den Kauf, Verkauf, das Senden von „crypto coins“, sowie zum Öffnen der Anwendung vor. Alternativ, etwa bei älteren Android-Versionen sind diese Handlungen auch über einen PIN möglich. Für Push-Benachrichtigungen wird der Dienst Firebase von Google verwendet und zur Absicherung der Kommunikation mit gewissen Schnitstellen wird „Certificate Pinning“ eingesetzt.
- Bei der statischen Analyse wurde eine unsichere Schlüsselableitungsfunktion in einer eingebundenen Komponente gefunden, die von einem Dritthersteller bereitgestellt wird.

## 4.1. Verwendung von Hardware-Schutzmechanismen

Sollte die App auf einem Android-Gerät betrieben werden, das zumindest Android 6 (API Level 23) Für die Generierung und Ablage von kryptographischen Schlüsseln wird auf die in Android inkludierte „Java Cryptographic Architecture“ (JCA) zurückgegriffen. Durch Angabe des KeyStore-Providers „AndroidKeyStore“ delegiert die Anwendung dem Betriebssystem die Verantwortung über die sichere Ablage. Es wurden im Kontext dieser Analyse keine Anzeichen dafür vorgefunden, dass die Anwendung sicherstellen würde, dass die Speicherung tatsächlich in einem Hardware- und nicht in einem Software-Schlüsselspeicher stattfindet. Letzteres würde über das Betriebssystem Für den Handel mit virtuellen Währungen, dem Senden von Coins an Kontakte, sowie zur Freigabe der Funktionalität nach Öffnen der Anwendung, kann wahlweise ein biometrischer Faktor verwendet werden. Alternativ wird auf einen PIN-Code zurückgegriffen, der bei der Einrichtung der Anwendung festgelegt werden muss und auch als zweiter Faktor für Aktionen dient, die andernfalls der Fingerabdruck erfüllt. Für die Implementierung der Fingerabdruck-Funktionalität wird auf eine von Google bereitgestellte Programmbibliothek zurückgegriffen [14] . Für gewisse Samsung-Geräte (z.B. Galaxy S8, S9, A20 und weitere) sind Ausnahmen bei der Verwendung hinterlegt, da es dem 14 https://developer.android.com/jetpack/androidx/releases/biometric

## 4.2. Push Notifications

Die Anwendung verwendet für Push Notifications den Dienst Firebase von Google und bezieht PushBenachrichtigungen von der Cloud-Datenbank https://bitx-wallet.firebaseio.com.

## 4.3. Kritische Berechtigungen

Die „Luno: Buy Bitcoin, Ethereum and Cryptocurrency“-App fordert bei Verwendung die Gewährung folgender „dangerous permissions“ an, die Zugriff auf persönliche Daten von BenutzerInnen oder sensible Geräteinformationen involvieren:

- *CAMERA*

- *GET_ACCOUNTS*
- *READ_CONTACTS* Ermöglicht der Applikation auf die gespeicherten Kontakte des Nutzers zuzugreifen.
- *READ_EXTERNAL_STORAGE* Ermöglicht es einer App von externem Speicher, etwa dem integrierten Massenspeicher oder
- RECORD_AUDIO Erlaubt es einer App das Mikrofon des Endgeräts zu aktivieren bzw. damit aufzuzeichnen.
- *WRITE_EXTERNAL_STORAGE* Ermöglicht es einer App auf externen Speicher, etwa den integrierten Massenspeicher oder

## 4.4. Verwendung externer Frameworks für „User Tracking“

Die Beschreibung der Anwendung auf Google Play verweist auf eine Datenschutzerklärung [15] . Darin wird beschrieben, welche Daten von der Webseite www.luno.com erhoben werden, insbesondere auch im Hinblick auf die Verwendung und das Tracking von User durch Cookies. Es findet sich in der aktuellen Fassung jedoch weder ein Verweis auf die Mobilanwendung, noch auf etwaige Im Code der Anwendung ist die Einbindung folgender Frameworks von Drittanbietern ersichtlich:

| Tracker | Zweck |
|---|---|
| Facebook Analytics | Analyse der App-Verwendung |
| Facebook Login | Login-Funktionalität via Facebook |
| Facebook Share | Teilen von Inhalten via Facebook |
| Google Analytics | Analyse der App-Verwendung |
| Google CrashLytics | Analyse und Meldung von Bugs |
| Google Firebase Analytics | Analyse der App-Verwendung |
| Google Tag Manager | Marketing-Optimierung |
| LeanPlum | Marketing-Optimierung |
| mParticle | Marketing-Optimierung |
| Singular | Marketing-Optimierung |

15 https://www.luno.com/legal/privacy

## 4.5. Netzwerkkommunikation

Neben den im vorigen Punkt angeführten Tracking-Diensten, kommuniziert die App vorwiegend mit https://api.mybitx.com. Für Zahlungen bzw. den Kauf virtueller Währungen wird eine Schnittstelle von https://checkout.com eingebunden. Um die Identität neuer BenutzerInnen zu verifizieren, inkludiert die

Anwendung den Dienst https://www.onfido.com. Diese Schnittstelle erfordert für die korrekte Funktionalität unter anderem die Berechtigungen CAMERA und die RECORD_AUDIO.

Das Wallet mit „crypto coins“ wird prinzipiell online über eine REST API verwaltet und ist mit einem Passwort geschützt, das von BenutzerInnen am Mobilgerät gesetzt bzw. geändert werden kann.

#### 4.5.1. Certificate Pinning

Die Anwendung inkludiert eine dedizierte Konfiguration [16], die Angaben zur Sicherheit von Netzwerkkommunikation trifft. Darin enthalten sind „Zertifikat-PINs“ für den https://checkout.com Dienst. Separat dazu prüft die Anwendung in einer eigenen Implementierung mittels Pinning auch Zugriffe auf die Domain https://api.mybitx.com sowie für Verbindung zum Google Messaging System, über das Push-Benachrichtigungen empfangen werden. Für weitere Domains bzw. Tracker-Dienste konnte keine Verwendung von Pinning festgestellt werden.

### 4.6. Verwendung von Kryptographie

Die in der “Luno: Buy Bitcoin, Ethereum and Cryptocurrency”-Anwendung enthaltene Programmbibliothek des Drittherstellers LeanPlum inkludiert Kryptographie zur verschlüsselten Ablage von Daten in den „SharedPreferences“.

Die Umsetzung dieser Verschlüsselung in der Klasse `com.leanplum.internal.AESCrypt` weist sicherheitskritische Mängel auf:

- Schlüsselableitung mittels PBKDF2

o Verwendung eines konstanten Passworts: `"Lq3fz" + appId + "bLti2"` wobei `appId` in der Datei `res/xml/strings.xml` hinterlegt ist als Zeichfolge:

```
      app_J1ExRRY0a98jcXHZf6ZaaKdQnoy0HAkHmBVdQTTZH3I
``` `o` Verwendung eines konstanten Salt-Wertes: `L3@nP1Vm` `o` 1000 Iterationen zur Ableitung eines AES-256-Schlüssels (unbedenklich)

   - Verschlüsselung mittels AES-256 bzw. AES/CBC/PKCS5Padding

`o` Verwendung des zuvor abgeleiteten Schlüssels `o` Konstanter Initialisierungsvektor (IV): `__l3anplum__iv__` Eine Programmbibliothek des Drittherstellers mparticle verwendet Kryptographie zur Berechnung eines HMAC-SHA256-Wertes. Im Zuge der in diesem Projekt vorgenommenen Analyse ließ sich jedoch nicht feststellen, in welchem Kontext bzw. mit welchem Passwort diese Berechnung passiert.

16 [https://developer.android.com/training/articles/security-config] (https://developer.android.com/training/articles/security-config)

## 5. App: eToro: Social-Trading

App selbst ist über das Cordova-Framework als Cross-Platform-Applikation realisiert worden.

|eToro: Social-Trading|Col2|
|---|---|
|**Link:**|Play Store|
|**Paket-Name:**|com.etoro.openbook|
|**Version:**|296.0.0|
|**Veröffentlichung:**|12.01.2021|
|**Downloads:**|5.000.000+|
|**Minimal unterstützte** **OS-Version:**| 5.1|

Die Anwendung unterstützt Android ab Version 5.1. Eine Authentifizierung, etwa über einen Fingerabdruck-Scanner oder der Einsatz von Hardware-Schutzmechanismen, ist nicht vorgesehen.

![[Platzhalter: Alt-Text]](abbildung-4.webp)

### 5.1. Push Notifications

Die Anwendung verwendet für Push Notifications den Dienst Firebase von Google und bezieht Push[Benachrichtigungen von der Cloud-Datenbank https://boxwood-harmony-102611.firebaseio.com.](https://boxwood-harmony-102611.firebaseio.com/)

### 5.2. Kritische Berechtigungen

Die „eToro: Social-Trading“-App fordert bei Verwendung die Gewährung folgender „dangerous permissions“ an, die Zugriff auf persönliche Daten oder sensible Geräteinformationen involvieren:

- _GET_ACCOUNTS_

- _READ_EXTERNAL_STORAGE_ Ermöglicht es einer App von externem Speicher, etwa dem integrierten Massenspeicher oder

- _READ_PHONE_STATE_ Ermöglicht den schreibgeschützten Zugriff auf den Telefonstatus, einschließlich der Telefonnummer des Geräts, aktueller Informationen zum Mobilfunknetz, des Status aller laufenden Anrufe und einer Liste aller auf dem Gerät registrierten Telefonnummern.

- _WRITE_EXTERNAL_STORAGE_ Ermöglicht es einer App auf externen Speicher, etwa den integrierten Massenspeicher oder

### 5.3. Verwendung externer Frameworks für „User Tracking“

Die Beschreibung der Anwendung auf Google Play verweist auf eine Datenschutzerklärung [17] . Darin [wird beschrieben, welche Daten von der Webseite www.etoro.com erhoben werden, insbesondere](http://www.etoro.com/) auch im Hinblick auf die Verwendung und das Tracking von User durch Cookies. Es findet sich in der aktuellen Fassung jedoch weder ein Verweis auf die Mobilanwendung, noch auf etwaige Im Code der Anwendung ist die Einbindung folgender Frameworks von Drittanbietern ersichtlich:

|Tracker|Zweck|
|---|---|
|AppsFlyer|Analyse der App-Verwendung|
|Facebook Analytics|Analyse der App-Verwendung|
|Facebook Login|Login-Funktionalität via Facebook|
|Facebook Places|Standortbezogene Informationen via Facebook|
|Facebook Share|Teilen von Inhalten via Facebook|
|Google AdMob|Werbung|
|Google Firebase Analytics|Analyse der App-Verwendung|
|Tencent Stats|Analyse der App-Verwendung|
|Urbanairship|Marketing-Optimierung|

### 5.4. Netzwerkkommunikation

Neben den im vorigen Punkt angeführten Tracking-Diensten, kommuniziert die App vorwiegend mit [https://www.etoro.com](https://www.etoro.com/) bzw. verwandte Domains desselben Anbieters wie etwa [https://accountstat.etoro.com oder](https://accountstat.etoro.com/) [https://api.etorostatic.com. Zahlungen werden über die](https://api.etorostatic.com/) [Kommunikation mit https://billing-pci.etoro.com](https://billing-pci.etoro.com/) abgewickelt.

Die App inkludiert keine dedizierte Konfiguration [18] mit Angaben zur Sicherheit der Kommunikation.

#### 5.4.1. Certificate Pinning

Die vorliegende Applikation sieht keine Verwendung bestimmter Zertifikate („Certificate Pinning“) für

### 5.5. Verwendung von Kryptographie

Die „eToro: Social-Trading”-Anwendung setzt Kryptographie für zur Speicherung von Userdaten in den sog. „SharedPreferences“ ein.

Als Cordova-Web-Anwendung greift sie zur Realisierung dieser Funktionalität auf die frei verfügbare Erweiterung „cordova-plugin-nativestorage“ [19] zurück. Die Implementierung enthält eine Schlüsselableitung mittels PBKDF2 mit 10.000 Iterationen, wobei das Passwort vom BenutzerIn bereitgestellt wird und der Salt-Wert dynamisch und kryptographisch sicher über die `SecureRandom` API generiert wird.

Wie im offiziellen Quellcode des Plugins gut nachvollziehbar, wird der kryptographische Schlüssel nach Ableitung geloggt [20] und somit „geleaked“. Dieses Sicherheitsproblem ist nicht zuletzt deshalb kritisch, weil es neben der „eToro: Social-Trading”-Anwendung auch alle weiteren Applikationen betrifft, die dieses Plugin verwenden.

17 [https://www.etoro.com/customer-service/privacy/]
(https://www.etoro.com/customer-service/privacy/) 18
[https://developer.android.com/training/articles/security-config]
(https://developer.android.com/training/articles/security-config) 19
[https://github.com/TheCocoaProject/cordova-plugin-nativestorage/]
(https://github.com/TheCocoaProject/cordova-plugin-nativestorage/) 20
https://github.com/TheCocoaProject/cordova-
plugin[nativestorage/blob/master/src/android/Crypto.java#L45]
(https://github.com/TheCocoaProject/cordova-plugin-
nativestorage/blob/master/src/android/Crypto.java#L45)

## 6. App: Bitcoin Wallet

Mobilgerät.

|Bitcoin Wallet|Col2|
|---|---|
|**Link:**|Play Store|
|**Paket-Name:**|de.schildbach.wallet|
|**Version:**|8.0.7|
|**Veröffentlichung:**|05.12.2020|
|**Downloads:**|5.000.000+|
|**Minimal unterstützte** **OS-Version:**| 6.0|

Version 6.0. Google schreibt Herstellern ab dieser Version vor, dass Hardware-

Verschlüsselung des Geräts standardmäßig aktiviert sein muss (siehe Android CDD [21], Abschnitt 9.9).

![[Platzhalter: Alt-Text]](abbildung-5.webp)

### 6.1. Verwendung von Hardware-Schutzmechanismen

Für die Generierung und Ablage von kryptographischen Schlüsseln wird auf die in Android inkludierte „Java Cryptographic Architecture“ (JCA) zurückgegriffen. Durch Angabe des KeyStore-Providers „AndroidKeyStore“ delegiert die Anwendung dem Betriebssystem die Verantwortung über die sichere Ablage. Es wurden im Kontext dieser Analyse keine Anzeichen dafür vorgefunden, dass die Anwendung sicherstellen würde, dass die Speicherung tatsächlich in einem Hardware- und nicht in einem Software-Schlüsselspeicher stattfindet. Da die App erst ab Android 6.0 betrieben werden kann und für Neugeräte ab dieser Version oben genannte Anforderung nach einem sicheren Für den Handel mit virtuellen Währungen, dem Senden von Coins an Kontakte, sowie zur Freigabe der Funktionalität nach Öffnen der Anwendung, kann ein PIN-Code gesetzt werden, der bei der Einrichtung der Anwendung festgelegt werden muss und auch als zweiter Faktor für Aktionen dient.

### 6.2. Push Notifications

Die Anwendung verwendet keine Push-Benachrichtigungen.

21 [https://source.android.com/compatibility/android-cdd.pdf] (https://source.android.com/compatibility/android-cdd.pdf)

### 6.3. Kritische Berechtigungen

Die „Bitcoin Wallet“-App fordert bei Verwendung die Gewährung folgender „dangerous permissions“ an, die Zugriff auf persönliche Daten von BenutzerInnen oder sensible Geräteinformationen involvieren:

- _CAMERA_

### 6.4. Verwendung externer Frameworks für „User Tracking“

Die vorliegende Anwendung verwendet keine Frameworks für „User Tracking“. Die im PlayStore angegebene Datenschutzerklärung [22] erläutert den Bedarf einzelner Berechtigungen.

### 6.5. Netzwerkkommunikation

Die „Bitcoin Wallet“ implementiert eine Peer-to-Peer-Kommunikation mit dem Bitcoin-Netzwerk und verwendet hier für die öffentlich verfügbare Programmbibliothek bitcoinj [23] .

Darüber hinaus findet Kommunikation nur mit (Sub-)Domains von

```
[https://wallet.schildbach.de,](https://wallet.schildbach.de/)
[https://coingecko.com und https://bitcoinaverage.com](https://coingecko.com/)
statt.

Die Anwendung inkludiert eine dedizierte Konfiguration [24], die Angaben zur
Sicherheit von Netzwerkkommunikation trifft. Darin ist explizit festgelegt, dass
Datenverkehr auch über HTTP stattfinden darf. Lediglich für die Kommunikation mit
oben angeführten Domains wird HTTPS explizit angefordert.

Die App implementiert in der Klasse
`de.schildbach.wallet.ui.send.RequestWalletBalanceTask` einen X509TrustManager zur
Validierung von Serverzertifikaten. Im vorliegenden Fall nimmt die Methode
`checkServerTrusted()` eine Kette von Zertifikaten entgegen. Der standardmäßig
gegebene Schutz durch die Validierung von Zertifikaten durch das Betriebssystem
wird dabei ausgeschalten und durch eine eigene Überprüfung ersetzt. Mangelnde
Kenntnis des konkreten Kontextes in dem diese Handlung passiert, verhindert eine
verlässliche Einschätzung der daraus resultierenden Folgen für die Sicherheit der
Anwendung.

#### 6.5.1. Certificate Pinning

Die vorliegende Applikation sieht keine Verwendung bestimmter Zertifikate
(„Certificate Pinning“) für

### 6.6. Verwendung von Kryptographie

Die “Bitcoin Wallet”-Anwendung verwendet Kryptographie im Zuge der bitcoinj-
Programmbibliothek zum Tausch von Bitcoins. Im Rahmen dieses Projekts lassen sich
keine Aussagen zur Sicherheit dieser Bibliothek treffen.

22 [https://github.com/bitcoin-wallet/bitcoin-wallet/wiki/PrivacyPolicy]
(https://github.com/bitcoin-wallet/bitcoin-wallet/wiki/PrivacyPolicy) 23
[https://github.com/bitcoinj/bitcoinj](https://github.com/bitcoinj/bitcoinj) 24
[https://developer.android.com/training/articles/security-config]
(https://developer.android.com/training/articles/security-config)

## 7. App: BUX X - Mobile Trading App

umgesetzt. Die gesamte Funktionalität ist in einer Anwendung enthalten.

|BUX X - Mobile Trading App|Col2|
|---|---|
|**Link:**|Play Store|
|**Paket-Name:**|com.getbux.android|
|**Version:**|2.39|
|**Veröffentlichung:**|25.01.2021|
|**Downloads:**|1.000.000+|
|**Minimal unterstützte** **OS-Version:**| 5.0|
```

Die vorliegende Anwendung unterstützt Android ab Version 5.0. Da die Schnittstelle zum Zugriff auf den Fingerabdruck-Scanner erst mit Android 6.0 eingeführt wurde, ist eine biometrische Authentifizierung auf älteren Geräten nicht möglich.



### 7.1. Verwendung von Hardware-Schutzmechanismen

Sollte die App auf einem Android-Gerät betrieben werden, das zumindest Android 6 (API Level 23) Für die Generierung und Ablage von kryptographischen Schlüsseln wird auf die in Android inkludierte „Java Cryptographic Architecture“ (JCA) zurückgegriffen. Durch Angabe des KeyStore-Providers „AndroidKeyStore“ delegiert die Anwendung dem Betriebssystem die Verantwortung über die sichere Ablage. Es wurden im Kontext dieser Analyse keine Anzeichen dafür vorgefunden, dass die Anwendung sicherstellen würde, dass die Speicherung tatsächlich in einem Hardware- und nicht in einem Software-Schlüsselspeicher stattfindet. Letzteres würde über das Betriebssystem Für den Handel mit virtuellen Währungen, dem Senden von Coins an Kontakte, sowie zur Freigabe der Funktionalität nach Öffnen der Anwendung, kann wahlweise ein biometrischer Faktor verwendet werden. Alternativ wird auf einen PIN-Code zurückgegriffen, der bei der Einrichtung der Anwendung festgelegt werden muss und auch als zweiter Faktor für Aktionen dient, die andernfalls der Fingerabdruck erfüllt. Für die Implementierung der Fingerabdruck-Funktionalität wird auf eine von Google bereitgestellte Programmbibliothek zurückgegriffen [25] . Für gewisse Samsung-Geräte (z.B. Galaxy S8, S9, A20 und weitere) sind Ausnahmen bei der Verwendung hinterlegt, da es dem 25
[https://developer.android.com/jetpack/androidx/releases/biometric]
(https://developer.android.com/jetpack/androidx/releases/biometric)

### 7.2. Push Notifications

Die Anwendung verwendet für Push Notifications den Dienst Firebase von Google und bezieht Push[Benachrichtigungen von der Cloud-Datenbank https://bux-io-bux-beta.firebaseio.com.](https://bux-io-bux-beta.firebaseio.com/)

### 7.3. Kritische Berechtigungen

Die „BUX X - Mobile Trading App“-App fordert bei Verwendung die Gewährung folgender „dangerous permissions“ an, die Zugriff auf persönliche Daten von BenutzerInnen oder sensible

- _CAMERA_

- _READ_EXTERNAL_STORAGE_ Ermöglicht es einer App von externem Speicher, etwa dem integrierten Massenspeicher oder

- _WRITE_EXTERNAL_STORAGE_ Ermöglicht es einer App auf externen Speicher, etwa den integrierten Massenspeicher oder

- RECORD_AUDIO Erlaubt es einer App das Mikrofon des Endgeräts zu aktivieren bzw. damit aufzuzeichnen.

### 7.4. Verwendung externer Frameworks für „User Tracking“

Die Beschreibung der Anwendung auf Google Play verweist auf eine Datenschutzerklärung [26] . Der Link führt zum Zeitpunkt dieser Analyse auf eine nicht existierende Seite (HTTP-Fehler 404).

Im Code der Anwendung ist die Einbindung folgender Frameworks von Drittanbietern ersichtlich:

|Tracker|Zweck|
|---|---|
|Adjust|Marketing-Optimierung|
|Facebook Analytics|Analyse der App-Verwendung|
|Facebook Login|Login-Funktionalität via Facebook|
|Google CrashLytics|Analyse und Meldung von Bugs|
|Google Firebase Analytics|Analyse der App-Verwendung|
|LeanPlum|Marketing-Optimierung|

Aus Datenschutzperspektive mitunter bedenklich ist der Einsatz der Dienste „Adjust“ und „LeanPlum“, die den Zweck verfolgen, Informationen über BenutzerInnen aus verschiedensten Quellen zu aggregieren und miteinander zu verknüpfen, um optimierte Marketing-Profile zu erstellen.

### 7.5. Netzwerkkommunikation

Neben den im vorigen Punkt angeführten Tracking-Diensten, kommuniziert die App vorwiegend mit [https://api.getbux.com. Um die Identität neuer BenutzerInnen zu verifizieren, inkludiert die](https://api.getbux.com/) Anwendung den Dienst [https://www.onfido.com. Diese Schnittstelle erfordert für die korrekte](https://www.onfido.com/) Funktionalität unter anderem die Berechtigungen CAMERA und die RECORD_AUDIO.

Das Wallet mit „crypto coins“ wird prinzipiell online über eine REST API verwaltet. Auch die Authentifizierung von Nutzer findet über eine gesonderte REST-Schnittstelle statt, die unter [https://auth.getbux.com erreichbar ist.](https://auth.getbux.com/) 26 [http://bux.ayondomarkets.com/en/legal-documents](http://bux.ayondomarkets.com/en/legal-documents)

#### 7.5.1. Certificate Pinning

Die Applikation sieht die Verwendung bestimmter Zertifikate („Certificate Pinning“) für eine Kommunikation vor. Die Anwendung inkludiert eine dedizierte Konfiguration [27], die Angaben zur Sicherheit von Netzwerkkommunikation trifft und in der die Hash-Werte von 8 public keys enthalten sind. In der Kommunikation mit Server wird

```
überprüft, ob zumindest einer dieser Werte in der präsentierten Zertifikatskette
vorkommt.

### 7.6. Verwendung von Kryptographie

Die in der “BUX X - Mobile Trading”-Anwendung enthaltene Programmbibliothek des
Drittherstellers LeanPlum inkludiert Kryptographie zur verschlüsselten Ablage von
Daten in den „SharedPreferences“.

Die Umsetzung dieser Verschlüsselung in der Klasse `com.leanplum.internal.AESCrypt`
weist sicherheitskritische Mängel auf:

   - Schlüsselableitung mittels PBKDF2

`o` Verwendung eines konstanten Passworts: `"Lq3fz" + appId + "bLti2"` wobei
`appId` in der Datei `res/xml/strings.xml` hinterlegt ist als Zeichfolge:
```

```
   app_J1ExRRY0a98jcXHZf6ZaaKdQnoy0HAkHmBVdQTTZH3I
```

``` o Verwendung eines konstanten Salt-Wertes: `L3@nP1Vm` o 1000 Iterationen zur Ableitung eines AES-256-Schlüssels (unbedenklich)

- Verschlüsselung mittels AES-256 bzw. AES/CBC/PKCS5Padding

o Verwendung des zuvor abgeleiteten Schlüssels o Konstanter Initialisierungsvektor (IV): `__l3anplum__iv__` Die Applikation setzt Kryptographie darüber hinaus ein, um einen vom User zu vergebenen PIN mittels Fingerabdruck zu schützen, sofern am Mobilgerät vorhanden. Dieser PIN wird folglich mittels AES/CBC/PKCS7Padding verschlüsselt und mitsamt dem (kryptographisch sicher generierten) Initialisierungsvektor in den „SharedPreferences“ der App abgelegt. Sollte am Mobilgerät kein Fingerabdruck hinterlegt sein, wird der PIN unverschlüsselt in den „SharedPreferences“ hinterlegt.

27 https://developer.android.com/training/articles/security-config

## 8. App: Abra Bitcoin Crypto Wallet - Buy Trade Earn Interest

Version 5.0. Eine Authentifizierung, etwa über einen Fingerabdruck-Scanner ist nicht vorgesehen.
```

| Abra Bitcoin Crypto Wallet | Col2 |
|---|---|
| **Link:** | Play Store |
| **Paket-Name:** | com.plutus.wallet |
| **Version:** | 90.0 |
| **Veröffentlichung:** | 19.01.2021 |
| **Downloads:** | 500.000+ |
| **Minimal unterstützte OS-Version:** | 5.0 |

**Executive Summary**

- Die App "Abra Bitcoin Crypto Wallet" verzeichnet laut Google Play mehr als 500.000 Installationen, ist für Geräte ab Android 5.0 verfügbar und verlangt bei Ausführung der jeweiligen Aktionen Zugriff auf die Kamera, den externen Speicher, Kontakte, den Standort des Mobilgeräts, sowie Accounts zu eingerichteten Diensten.
- Die App sieht keine Verwendung eines Fingerabdruck-Scanners für den Kauf, Verkauf, das Senden von „crypto coins", sowie zum Öffnen der Anwendung vor. Zum Öffnen der App kann ein PIN hinterlegt werden. Für Push-Benachrichtigungen wird der Dienst Firebase von Google verwendet.
- Die Anwendung weist Sicherheitsmängel bei der sicheren Netzwerkkommunikation auf bzw. wird (vorsätzlich) beliebigen, am Geräten installierten Zertifikatsausstellern vertraut.
- Die „Abra Bitcoin Crypto Wallet"-App verwendet eine Funktion zur Schlüsselableitung, die korrekt implementiert wurde.

## 8.1. Verwendung von Hardware-Schutzmechanismen

Für die Generierung und Ablage von kryptographischen Schlüsseln wird auf die in Android inkludierte „Java Cryptographic Architecture" (JCA) zurückgegriffen. Durch Angabe des KeyStore-Providers „AndroidKeyStore" delegiert die Anwendung dem Betriebssystem die Verantwortung über die sichere Ablage. Es wurden im Kontext dieser Analyse keine Anzeichen dafür vorgefunden, dass die Anwendung sicherstellen würde, dass die Speicherung tatsächlich in einem Hardware- und nicht in einem Software-Schlüsselspeicher stattfindet. Letzteres würde über das Betriebssystem Für den Handel mit virtuellen Währungen sowie zur Freigabe der Funktionalität nach Öffnen der Anwendung, kann auf einen PIN-Code zurückgegriffen werden, der bei der Einrichtung der Anwendung festgelegt werden muss und auch als zweiter Faktor für Aktionen dient. Dieser PIN wird prinzipiell in den „SharedPreferences" der Anwendung gespeichert.

## 8.2. Push Notifications

Die Anwendung verwendet für Push Notifications den Dienst Firebase von Google und bezieht PushBenachrichtigungen von der Cloud-Datenbank https://abraprod-992.firebaseio.com.

## 8.3. Kritische Berechtigungen

Die “Abra Bitcoin Crypto Wallet“-App fordert bei Verwendung die Gewährung folgender „dangerous permissions“ an, die Zugriff auf persönliche Daten von BenutzerInnen oder sensible

- ACCESS_COARSE_LOCATION Benötigt, um auf den ungefähren Standort des Geräts zuzugreifen.
- ACCESS_FINE_LOCATION Erforderlich, um auf den exakten Standort des Geräts zuzugreifen.
- CALL_PHONE Die App kann einen Telefonanruf ohne Nutzerbestätigung starten.
- *CAMERA*
- *GET_ACCOUNTS*
- *READ_CONTACTS* Ermöglicht der Applikation auf die gespeicherten Kontakte des Nutzers zuzugreifen.
- *READ_EXTERNAL_STORAGE* Ermöglicht es einer App von externem Speicher, etwa dem integrierten Massenspeicher oder
- *READ_PHONE_STATE* Ermöglicht den schreibgeschützten Zugriff auf den Telefonstatus, einschließlich der Telefonnummer des Geräts, aktueller Informationen zum Mobilfunknetz, des Status aller laufenden Anrufe und einer Liste aller auf dem Gerät registrierten Telefonnummern.
- *WRITE_EXTERNAL_STORAGE* Ermöglicht es einer App auf externen Speicher, etwa den integrierten Massenspeicher oder

## 8.4. Verwendung externer Frameworks für „User Tracking“

Die Beschreibung der Anwendung auf Google Play verweist auf eine Datenschutzerklärung [28] . Darin wird beschrieben, welche Daten von der Webseite www.abra.com erhoben werden, insbesondere auch im Hinblick auf die Verwendung und das Tracking von User durch Cookies. Es findet sich in der aktuellen Fassung jedoch weder ein Verweis auf die Mobilanwendung, noch auf etwaige Aus Datenschutzperspektive mitunter bedenklich ist der Einsatz des Dienstes „Branch“, der den Zweck verfolgt, Informationen über BenutzerInnen aus verschiedensten Quellen zu aggregieren und miteinander zu verknüpfen, um optimierte Marketing-Profile daraus zu erstellen. Diese Daten umfassen unter anderem die IP-Adresse, einen Fingerprint des aktuellen Geräts, sowie weitere gerätespezifische Merkmale, die an den Dienstleister gesendet werden.

Im Code der Anwendung ist die Einbindung folgender Frameworks von Drittanbietern ersichtlich:

28 https://www.abra.com/privacy-policy/

| Tracker | Zweck |
|---|---|
| AppsFlyer | Analyse der App-Verwendung |
| Apptentive | Marketing-Optimierung |
| Branch | Analyse der App-Verwendung |
| Facebook Analytics | Analyse der App-Verwendung |
| Facebook Login | Login-Funktionalität via Facebook |
| Facebook Places | Standortbezogene Informationen via Facebook |
| Facebook Share | Teilen von Inhalten via Facebook |
| Google CrashLytics | Analyse und Meldung von Bugs |
| Google Firebase Analytics | Analyse der App-Verwendung |
| mParticle | Marketing-Optimierung |

## 8.5. Netzwerkkommunikation

Neben den im vorigen Punkt angeführten Tracking-Diensten, kommuniziert die App vorwiegend mit https://www.abra.com bzw. zugehörigen Subdomains. Die Anwendung inkludiert eine dedizierte Konfiguration [29], die Angaben zur Sicherheit von Netzwerkkommunikation trifft. Darin ist explizit festgelegt, dass die Anwendung auf dem System manuell installierten User-Zertifikaten vertrauen soll, wenn sie im Debug-Modus [30] betrieben wird. Da dies nicht dem gewöhnlichen Anwendungsszenario entspricht, ist diese Konfiguration nicht per se problematisch.

Die App implementiert in der Klasse `com.plutus.wallet.util.bh` einen HostnameVerifier zur Validierung, ob der aufgerufene Hostname mit dem tatsächlichen übereinstimmt. Im vorliegenden Fall wird die standardmäßig sichere Prüfung durch eine eigene, unsichere überschrieben und folglich jeder Hostname akzeptiert. Im Allgemeinen ist diese Praxis problematisch, da die Anwendung z.B. bei Aufruf von google.at auch ein gültiges Zertifikat einer anderen Domain akzeptieren würde. Mangelnde Kenntnis des konkreten Kontextes in dem diese Handlung passiert, verhindert eine verlässliche Einschätzung der daraus resultierenden Folgen für die Sicherheit der Anwendung.

### 8.5.1. Certificate Pinning

Die vorliegende Applikation sieht keine Verwendung bestimmter Zertifikate („Certificate Pinning“) für

## 8.6. Verwendung von Kryptographie

Die “Abra Bitcoin Crypto Wallet“-Anwendung implementiert eine PBKDF2-Schlüsselableitung und Ver- bzw. Entschlüsselung mittels AES im ECB-Modus.

usgangspunkt ist hierbei ein vom User eingegebener PIN, der zusammen mit einem kryptographisch sicher generiertem Salt-Wert (Verwendung der `SecureRandom` API) in 1000 Iterationen verarbeitet wird, um einen Schlüssel für AES-256 abzuleiten. Ein Wert von zumindest 1000 Iterationen gilt gemäß PKCS#5-Spezifikation Version 2.1 (RFC 8018 [31] ) von 2017 als sicher. Aufgrund der starken Verschleierung von Variablen und des Datenflusses konnte im Rahmen dieser Analyse nicht festgestellt werden, in welchem Kontext diese Funktionalität eingesetzt wird.

29 https://developer.android.com/training/articles/security-config 30 https://developer.android.com/training/articles/security-config#debug-overrides 31 https://tools.ietf.org/html/rfc8018

## 9. App: CEX.IO Cryptocurrency Exchange

Version 5.0. Da die Schnittstelle zum Zugriff auf den

| CEX.IO Cryptocurrency Exchange | Col2 |
|---|---|
| **Link:** | Play Store |
| **Paket-Name:** | io.cex.app.prod |
| **Version:** | 5.23.1 |
| **Veröffentlichung:** | 29.01.2021 |
| **Downloads:** | 500.000+ |
| **Minimal unterstützte OS-Version:** | 5.0 |

Fingerabdruck-Scanner erst mit Android 6.0 eingeführt wurde, ist eine biometrische Authentifizierung auf älteren Geräten nicht möglich.

**Executive Summary**

- Die App „CEX.IO Cryptocurrency Exchange“ verzeichnet laut Google Play mehr als 500.000 Installationen, ist für Geräte ab Android 5.0 verfügbar und verlangt bei Ausführung der jeweiligen Aktionen Zugriff auf die Kamera, den externen Speicher und Accounts zu eingerichteten Diensten.
- Die App sieht die Verwendung eines Fingerabdruck-Scanners nur zum Öffnen der Anwendung vor. Alternativ sind diese Handlungen über einen PIN möglich. Für Push-Benachrichtigungen wird der Dienst Firebase von Google verwendet.
- Die Anwendung verwendet kein „Certificate Pinning“ zur Absicherung der Kommunikation.
- Bei der statischen Analyse wurden keine wesentlichen Implementierungsschwächen festgestellt.

## 9.1. Verwendung von Hardware-Schutzmechanismen

Sollte die App auf einem Android-Gerät betrieben werden, das zumindest Android 6 (API Level 23) Für die Generierung und Ablage von kryptographischen Schlüsseln wird auf die in Android inkludierte „Java Cryptographic Architecture“ (JCA) zurückgegriffen. Durch Angabe des KeyStore-Providers „AndroidKeyStore“ delegiert die Anwendung dem Betriebssystem die Verantwortung über die sichere Ablage. Es wurden im Kontext dieser Analyse keine Anzeichen dafür vorgefunden, dass die Anwendung sicherstellen würde, dass die Speicherung tatsächlich in einem Hardware- und nicht in einem Software-Schlüsselspeicher stattfindet. Letzteres würde über das Betriebssystem Zur Freigabe der Funktionalität nach Öffnen der Anwendung, kann wahlweise ein biometrischer Faktor oder PIN verwendet werden. Für die Implementierung der Fingerabdruck-Funktionalität wird auf die öffentlich verfügbare Programmbibliothek RxFingerprint [32] zurückgegriffen.

## 9.2. Push Notifications

Die Anwendung verwendet für Benachrichtigungen den Firebase-Dienst von Google und bezieht Push Notifications von der Cloud-Datenbank https://project-931427959788298615.firebaseio.com. 32 https://github.com/Mauin/RxFingerprint

## 9.3. Kritische Berechtigungen

Die „CEX.IO Cryptocurrency Exchange“-App fordert bei Verwendung die Gewährung folgender „dangerous permissions“ an, die Zugriff auf persönliche Daten von BenutzerInnen oder sensible

- *CAMERA*
- *GET_ACCOUNTS*
- *READ_EXTERNAL_STORAGE* Ermöglicht es einer App von externem Speicher, etwa dem integrierten Massenspeicher oder
- *WRITE_EXTERNAL_STORAGE* Ermöglicht es einer App auf externen Speicher, etwa den integrierten Massenspeicher oder

## 9.4. Verwendung externer Frameworks für „User Tracking“

Die Beschreibung der Anwendung auf Google Play verweist auf eine Datenschutzerklärung [33] . Darin wird beschrieben, welche Daten von der Webseite www.cex.io erhoben werden, insbesondere auch im Hinblick auf die Verwendung und das Tracking von User durch Cookies. Es findet sich in der aktuellen Fassung jedoch weder ein Verweis auf die Mobilanwendung, noch auf etwaige Drittdienste, die in die Mobilanwendung eingebunden sind und Daten verarbeiten.

Im Code der Anwendung ist die Einbindung folgender Frameworks von Drittanbietern ersichtlich:

| Tracker | Zweck |
|---|---|
| Google Analytics | Analyse der App-Verwendung |
| Google CrashLytics | Analyse und Meldung von Bugs |
| Google Firebase Analytics | Analyse der App-Verwendung |
| Google Tag Manager | Marketing-Optimierung |
| MixPanel | Marketing-Optimierung |

### 9.5. Netzwerkkommunikation

Neben den im vorigen Punkt angeführten Tracking-Diensten, kommuniziert die App vorwiegend mit https://cex.io. Die Anwendung inkludiert darüber hinaus Funktionalität, um via WebSockets Verbindungen zu Server herzustellen.

Die App inkludiert keine dedizierte Konfiguration [34] mit Angaben zur Sicherheit der Kommunikation.

Die „CEX.IO“-App implementiert in der Klasse `io.cex.cex.network.TrustedHostnameVerifier` einen HostnameVerifier zur Validierung, ob der aufgerufene Hostname mit dem tatsächlichen übereinstimmt. Im vorliegenden Fall wird die standardmäßig sichere Prüfung durch eine eigene, unsichere überschrieben und folglich jeder Hostname akzeptiert. Im Allgemeinen ist diese Praxis problematisch, da die Anwendung z.B. bei Aufruf von google.at auch ein gültiges Zertifikat einer anderen Domain akzeptieren würde. Mangelnde Kenntnis des konkreten Kontextes in dem diese Handlung passiert, verhindert eine verlässliche Einschätzung der daraus resultierenden Folgen für die Sicherheit der Anwendung.

33 https://cex.io/privacy 34 https://developer.android.com/training/articles/security-config

#### 9.5.1. Certificate Pinning

Die vorliegende Applikation sieht keine Verwendung bestimmter Zertifikate („Certificate Pinning“) für

### 9.6. Verwendung von Kryptographie

Die “CEX.IO Cryptocurrency Exchange”-Anwendung setzt Kryptographie nur beim Öffnen der Anwendung durch Verwendung der Programmbibliothek RxFingerprint ein. Im Rahmen dieses Projekts lassen sich keine Aussagen zur Sicherheit dieser Bibliothek treffen.

## 10. Fazit

Während etwa Banken im Umgang mit traditionellen Finanzinstrumenten legislativ und technisch umfassenden Sicherheitsvorgaben entsprechen müssen, haben Anbieter von Mobilanwendungen, die den Besitz und Handel mit Krypto-Währungen erlauben, keinerlei Vorschriften zu erfüllen. Wenngleich der gesellschaftliche Status und die praktische Relevanz von „crypto coins“ nicht mit realem Vermögen vergleichbar sein mögen, steckt dennoch sehr oft ein mit echtem Geld messbarer Wert in „Crypto-Wallets“.

Nicht zuletzt dieser Umstand hat in der jüngeren Vergangenheit auch bereits mehrfach dazu beigetragen, dass Vermögen aus Wallets gestohlen bzw. mit den persönlichen Daten realer Personen entsprechende An- oder Verkäufe von Coins getätigt wurden.

Im Zuge dieses Projekts wurde der Fokus auf Sicherheitsaspekte in vergleichsweise beliebten Android-Anwendungen gelegt, die den Handel bzw. die Verwaltung von virtuellen Währungen über das Mobilgerät ermöglichen. Das Ziel bestand darin, einen Einblick zu gewinnen, welches Sicherheitsniveau verbreitete Anwendungen bieten, wenngleich es hierfür keine rechtlich bindenden Vorschriften gibt, denen die Anbieter Folge zu leisten hätten.

Die in diesem Projekt gewonnenen Ergebnisse der durchgeführten Sicherheitsanalysen zeigen im Wesentlichen ein zufriedenstellendes Gesamtbild. In keiner der untersuchten Anwendungen weist die Hauptfunktionalität eklatante Sicherheitsmängel auf. Dennoch lassen sich in nahezu jeder App Implementierungsmuster beobachten, die die eigentlich vom Hersteller beabsichtigte Sicherheit schwächen. In mehreren Fällen passiert dies durch die Verwendung von Programmbibliotheken Dritter, die ihrerseits Sicherheitsmängel in eigentlich korrekt umgesetzte Apps einfügen. Ebenfalls negativ fällt der weit verbreitete Einsatz von Tracker-Frameworks auf. In nur einem Fall hat die der App beigefügte Datenschutzerklärung des Herstellers die Verwendung persönlicher Daten durch diese Frameworks überhaupt thematisiert.